

Taller de <?XML>

SABER-ULA

Hilda Contreras hyelitza@ula.ve

3er Evento “Teleinformación e Innovación”

Nov. 2004

Contenido

1. Introducción
2. Preparándonos para trabajar con XML
3. Documentos XML bien formados
4. Documentos XML válidos
5. Consultando documentos XML
6. Transformando XML
7. Uso de XML en Bases de Información

1 - Introducción

1. Historia
2. Qué es XML? y Qué NO es XML?
3. Porqué XML?
4. Conceptos básicos
5. Ejemplo

1.1. Historia

- Internet (1984)
- SGML (1986) gestación desde principios de los años 70
- Multimedia (1986)
- PDF (1992)
- NSF DLI National Science Foundation's Digital Libraries Initiative (1994)
- World Wide Web Consortium - **W3C** (1994)
- WWW (1994)
- XML (1998)

1.1. Historia

World Wide Web Consortium - W3C (1994)

<http://www.w3c.org>

- Constituido con el objetivo de desarrollar protocolos comunes para Internet
- Consorcio de industrias internacionales: MIT (EEUU), INRIA (Francia) y Keio University (Japón)
- Soporte oficial del DARPA (EEUU) y La Comisión Europea

1.2. Qué es XML?

- Es un subconjunto de SGML (Standard Generalized Markup Language) simplificado y adaptado a Internet
- Extensible Markup Language (XML), es más que un lenguaje de **marcado**
- Es un **meta-lenguaje** es un lenguaje para definir lenguajes

1.2. Qué es XML?

“Lenguaje de **marcado**” basado en marcas:

- **Marcas** [mark-up] construcciones con etiquetas. Comienzan con “<” y terminan con “>”.
- **Datos** resto de contenido del documento que se encuentra entre marcas
- Ejemplo:

<autor>Deepak Chopra</autor>

1.2. Qué es XML?

“Meta-lenguaje”:

Información (marca): Deepak Chopra

Meta-información (dato): <autor>

- Describir otros lenguajes
- Crear etiquetas propias

1.2. Qué NO es XML?

- NO es una versión mejorada de HTML
- NO es un lenguaje para hacer mejores páginas Web
- NO es un lenguaje sustituto de HTML
- NO es un lenguaje difícil

1.3. Por qué XML?

- Es un estándar internacional reconocido por W3C (1998)
- Su utilización es libre
- Permite la utilización de múltiples alfabetos en diferentes plataformas
- Fácil procesamiento
- Separa el contenido de los datos de su presentación

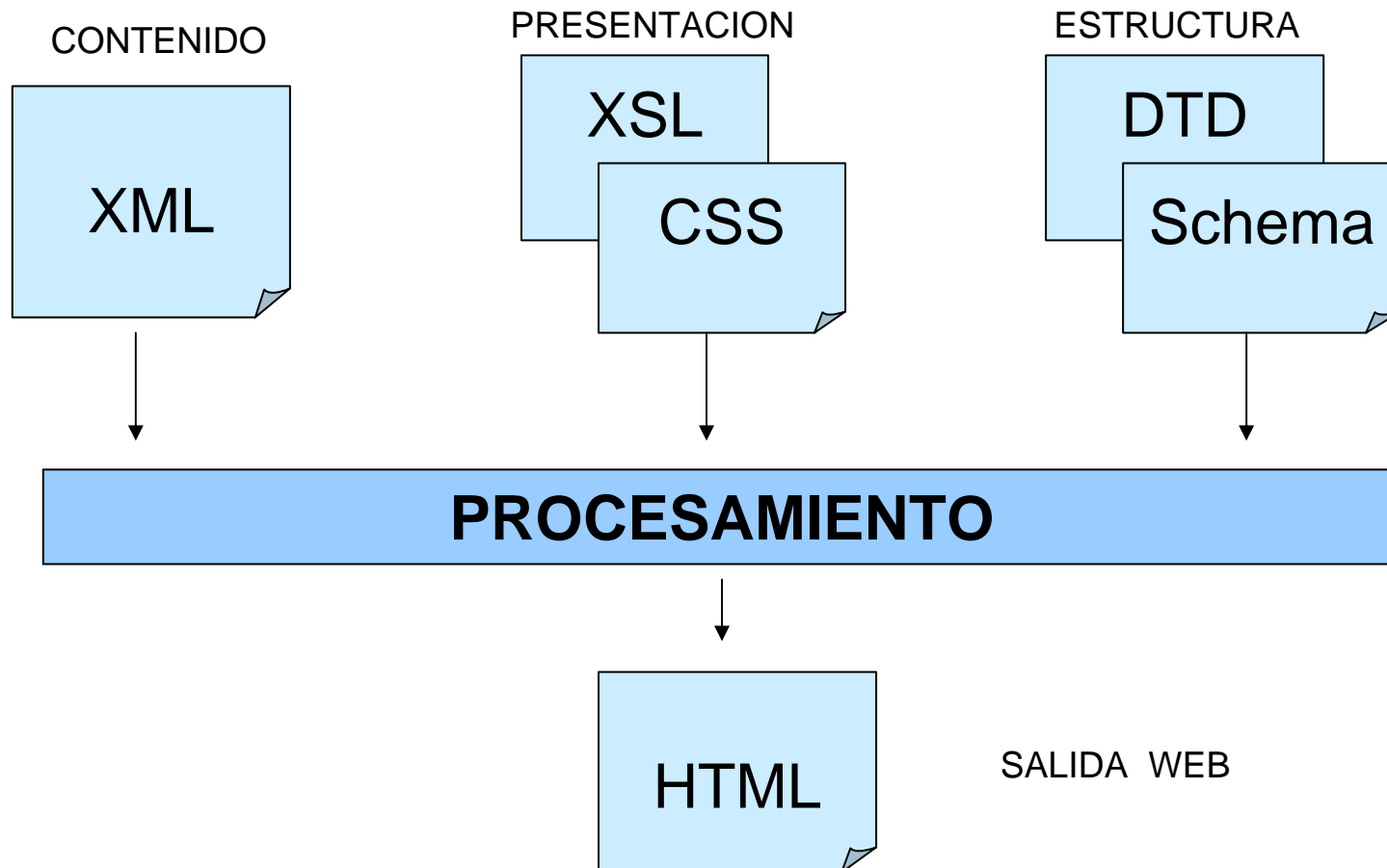
1.4. Conceptos básicos

Separación lógica de **contenido**, descripción de la **estructura** y **presentación**. Esto permite máxima independencia y flexibilidad.

- *Contenido*: datos. Documento XML.
- *Estructura*: reglas de estructura de los datos. DTD y XML Schema.
- *Presentación*: el formato para mostrar la información. Hojas de estilo

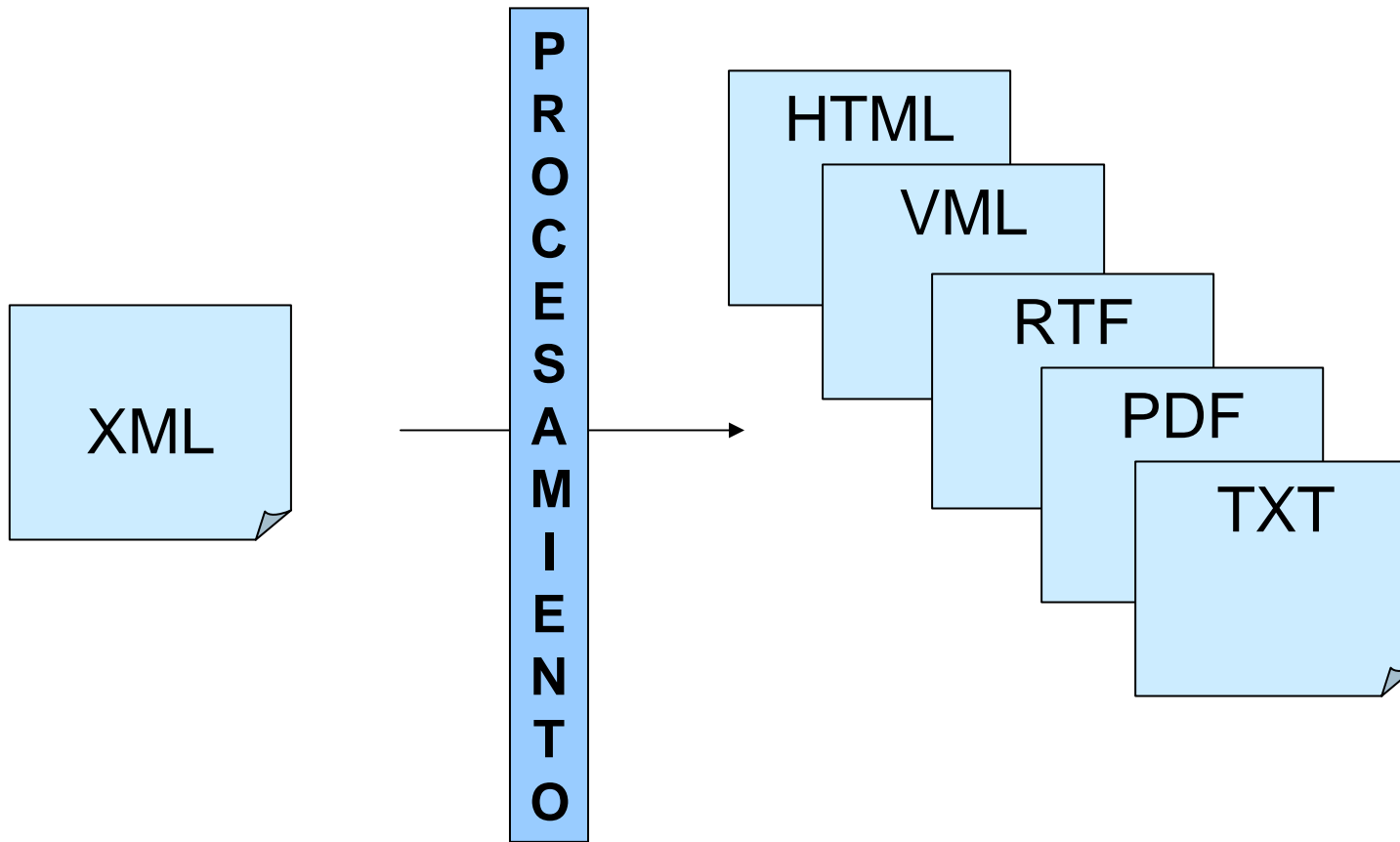
1.4. Conceptos básicos

Separación de procesamiento, presentación, estructura y contenido:



1.4. Conceptos básicos

Muchas formas de presentación a partir de un documento XML:



1.4. Conceptos básicos

Sopa de letras:

XML	WWW	PHP	ASCII
XSL	W3C	MSXSL	XQuery
DTD	SGWL	MSXML	XSLT
XSD	PDF	CDATA	SOAP
Schema	VML	PCDATA	XHTML
HTML	TXT	UTF-8	API
RDF	DOM	Xpath	
CSS	SAX	ISO	

1.5. Ejemplo

- Un Ejemplo de HTML:

Deepak Chopra

<i>El sendero del Mago</i>

precio: Bs. 30.000

1.5. Ejemplo

- El mismo Ejemplo en XML:

<libro>

<autor>Deepak Chopra</autor>

<titulo>El sendero del Mago</titulo>

<precio moneda="bolivares">30.000</precio>

</libro>

Comentar las diferencias!.

2 - Preparándonos para trabajar con XML

1. Qué es un archivo XML?
2. Herramientas necesarias
3. Editando un archivo XML
4. Visualizando un archivo XML

PRÁCTICA #1: Edición y visualización de un archivo XML

2.1. Qué es un archivo XML?

- Un archivo de texto ASCII
- Un archivo con extensión “.xml”
- Un archivo cuyo contenido sigue las reglas sintácticas de XML



2.2. Herramientas necesarias

- Un Editor de Texto
- Un Explorador o Navegador de Internet (*)
- Un herramienta de procesamiento y transformación para XML o parser (*)

2.2. Herramientas necesarias

Parser: Analizador sintáctico

Procesa el contenido de un archivo XML para:

- Validar
- Transformar

2.2. Herramientas necesarias

Parser

- Pueden incluir validación o no
- Pueden realizar transformaciones o no
- Pueden exponer la información de diferentes formas (DOM, SAX)
- Existen para la mayoría de lenguajes y plataformas de desarrollo (VB, Php, Perl, Java, etc.)

2.2. Herramientas necesarias

Lista de algunos Parser disponibles:

- Xerces (Apache)
- XML4J (IBM)
- Crimson (Apache)
- Project X (Sun Microsystems)
- MSXML (Microsoft)
- XP (James Clark)
- Ælfred (Microstar Software)
- Lark/Larval (Tim Bray)
- XJ (Data Channel)

2.2. Herramientas necesarias

Usaremos el parser de Microsoft MSXML.dll

- Realiza validaciones contra DTD
- Realiza transformaciones con soporte XSL
- Gratuito y redistribuible libremente

2.3 – Editando un archivo XML

PRÁCTICA #1:

- Ejecutar cualquier editor de texto
- Escribir este texto:

```
<libro>  
<autor>Deepak Chopra</autor>  
<titulo>El sendero del Mago</titulo>  
<precio moneda="bolivares">30.000</precio>  
</libro>
```
- Guardar como “libro1.xml” en tallerxml

2.4 – Visualización de un archivo XML

PRÁCTICA #1:

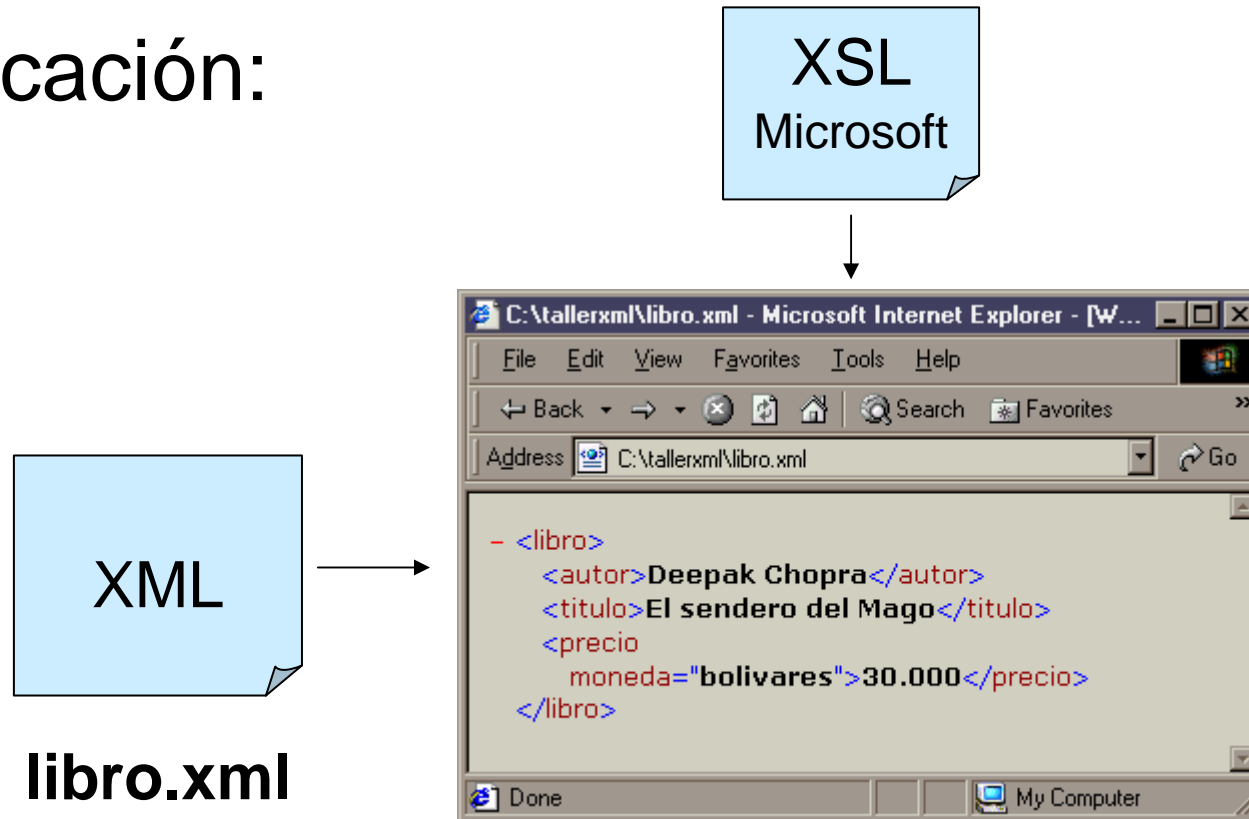
- Ejecutar un navegador de Internet, por ejemplo IE (Internet Explorer)
- Abrir el archivo libro1.xml desde el directorio tallerxml

Opciones: File – Open – Browse

- Observar la forma en que se muestra el archivo XML en el navegador.

PRÁCTICA #1

Explicación:



3 - Documentos XML bien formados

1. Reglas sintácticas XML
2. Documentos bien formados
3. Espacios de nombres XML (namespaces)

PRÁCTICA #2: Documentos XML bien formados

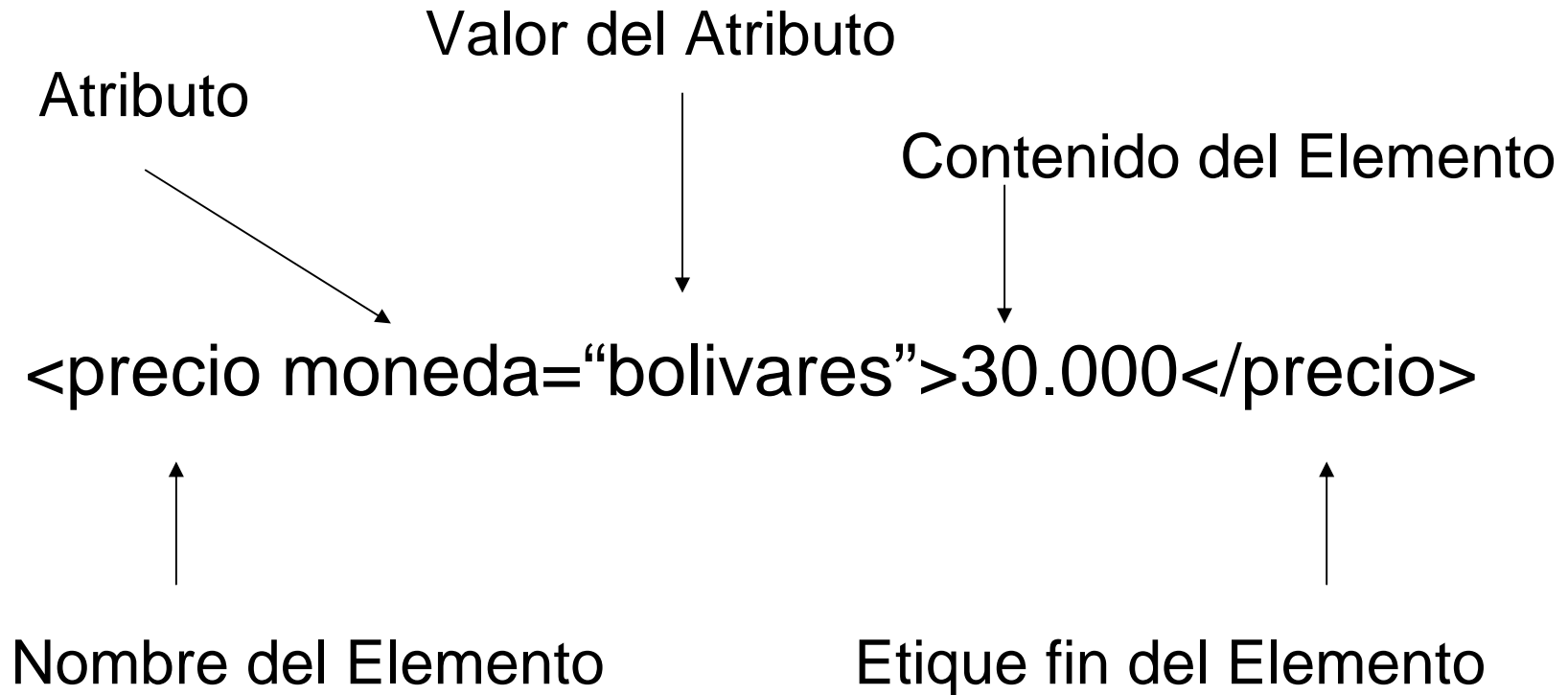
3.1. Reglas sintácticas XML

Cómo debo escribir los **Elementos**?

- Siempre comienzan con una <etiqueta>
- Siempre termina con una </etiqueta> del mismo nombre
- Elemento vacío <etiqueta/>
- Es “case-sensitive”, es decir <autor> no es igual a <Autor>
- No se permiten espacios en blanco ni saltos de línea en el nombre

3.1. Reglas sintácticas XML

- **Atributo**



3.1. Reglas sintácticas XML

Cómo debo escribir los **Atributos**?

- Los atributos son un par de:
nombre_atributo="valor atributo"
- Los valores de los atributos siempre deben estar entre comillas simples (') o dobles (").
- Si se permiten espacios en blanco en el valor del atributo.

3.1. Reglas sintácticas XML

Cómo debo escribir los **nombres** de atributos y elementos?

- Empezar por una letra
- Continuar con letras, dígitos, guiones, rayas, puntos o dos puntos.
- No se permiten espacios en blanco
- No usar la palabra “XML” como comienzo de un nombre

3.1. Reglas sintácticas XML

Cómo debo escribir el contenido de la información?:

- No usar **entidades predefinidas**
- Usar el conjunto de caracteres según la codificación especificada (encoding).
- Si deben usarse **entidades predefinidas** identificar como **CDATA**

3.1. Reglas sintácticas XML

Entidades Predefinidas: XML 1.0 define 5 entidades de caracteres especiales:

<	<
>	>
&	&
'	'
“	"

<temperatura>< 0</temperatura>

3.1. Reglas sintácticas XML

Sección CDATA (Character Data)

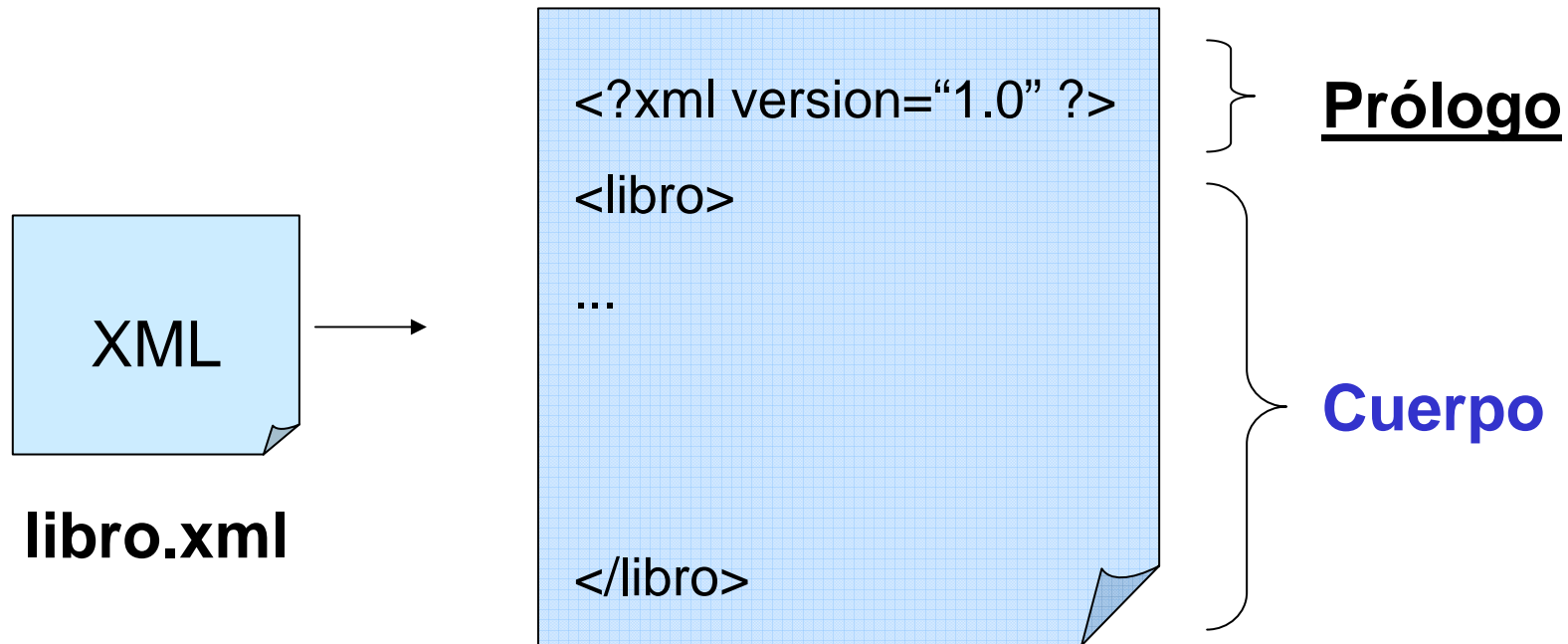
- Comienza con `<![CDATA[` y termina con `]]>`
- Puede contener cualquier caracteres incluso entidades predefinidas sin codificar, excepto la cadena de cierre `]]>`
- Uso insertar HTML o Javascript

```
<temperatura>&lt; 0</temperatura>
```

```
<temperatura><![CDATA[<0]]></temperatura>
```

3.1. Reglas sintácticas XML

- Estructura



3.1. Reglas sintácticas XML

Cómo debo escribir el **Prólogo**?

- El prólogo es opcional
- La primera línea debe comenzar con `<?xml` y terminar con `?>`.
- La primera línea especifica la **versión** y la **codificación** de caracteres

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

- La segunda línea especifica documentos asociados (DTD, Schema, XSL)

3.1. Reglas sintácticas XML

Ejemplos de prólogos:

```
<?xml version="1.0" encoding="ISO-8859-7"?>
```

```
<?xml version="1.0" encoding="UTF-16" standalone="yes"?>
```

```
<?xml version="1.0" encoding="Big-5" standalone="yes"?>
```

```
<!DOCTYPE clima SYSTEM "clima.dtd">
```

```
<?xml version="1.0">
```

```
<?xml:stylesheet type="text/xsl" href="prueba.xls"?>
```

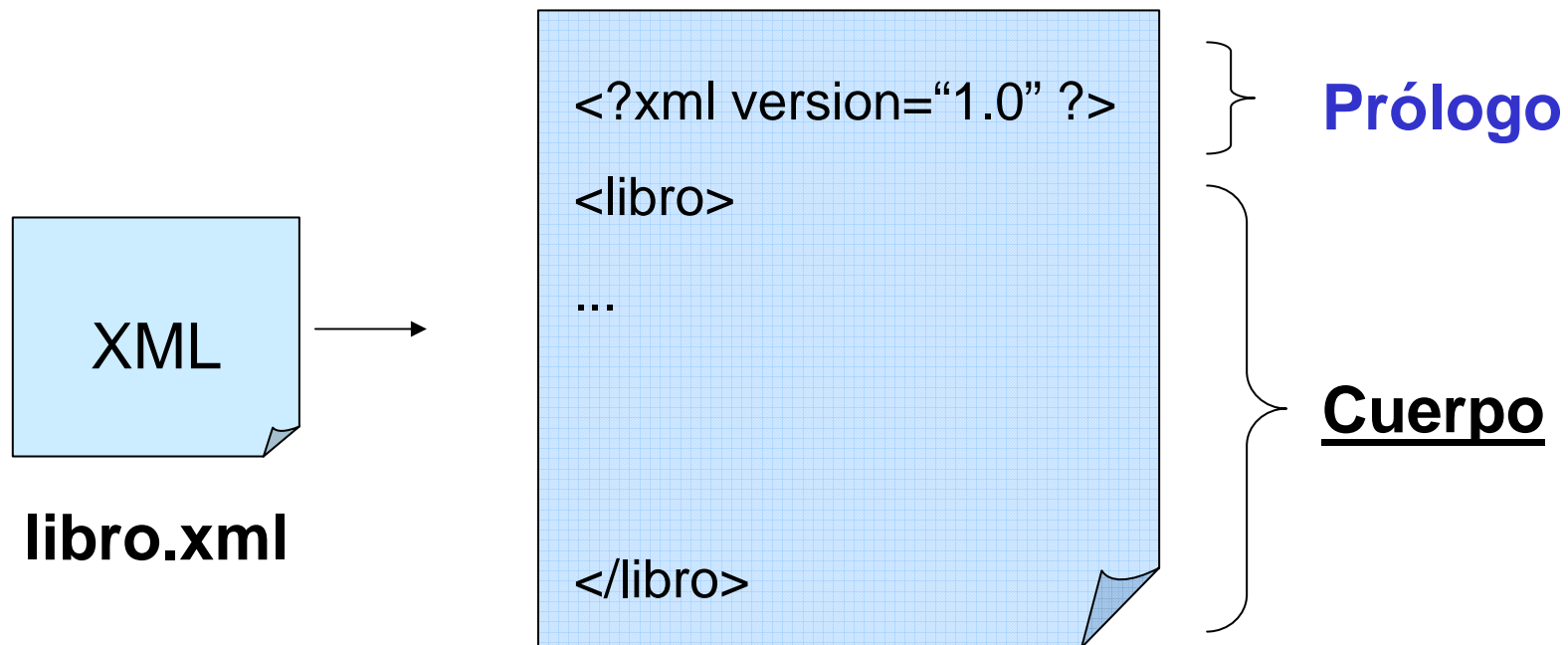
3.1. Reglas sintácticas XML

Codificación:

- Codificación de caracteres usada en el archivo XML
- Identificación del lenguaje basado en ISO o Unicode
- Valor implícito por defecto Unicode comprimido: `encoding="UTF-8"`

3.1. Reglas sintácticas XML

- Estructura



3.1. Reglas sintácticas XML

El cuerpo de un archivo XML tiene una estructura **jerárquica**:

- Estructura de árbol correctamente anidados
- No se pueden superponer elementos
- Debe haber un elemento raíz por documento XML
- Todas las etiquetas deben cerrarse

3.1. Reglas sintácticas XML

- Incorrectamente anidado

```
<libro>  
<autor>Deepak Chopra  
<titulo>El sendero del Mago</autor></libro></titulo>
```

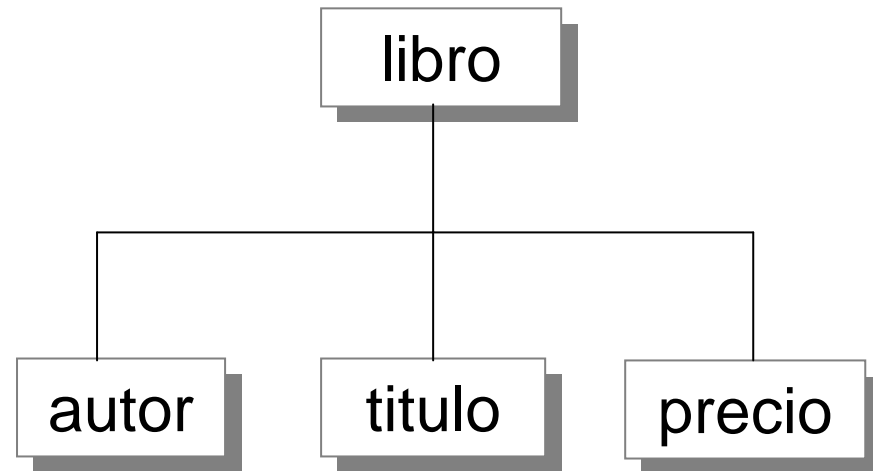
- Correctamente anidado

```
<libro>  
<autor>Deepak Chopra</autor>  
<titulo>El sendero del Mago</titulo>  
</libro>
```

3.1. Reglas sintácticas XML

Estructura Jerárquica de Árbol

```
<libro>  
<autor>Deepak Chopra</autor>  
<titulo>El sendero del  
Mago</titulo>  
<precio  
moneda="bolivares">30.000</pr  
ecio>  
</libro>
```



3.1. Reglas sintácticas XML

- Una pregunta antes de continuar:
Cuándo un dato es atributo o elemento?

```
<libro>  
<autor>Deepak Chopra</autor>  
<titulo>El sendero del  
Mago</titulo>  
<precio  
moneda="bolivares">30.000</pr  
ecio>  
</libro>
```

```
<libro>  
<autor>Deepak Chopra</autor>  
<titulo>El sendero del  
Mago</titulo>  
<precio>30.000</precio>  
<moneda>bolivares</moneda>  
</libro>
```

3.1. Reglas sintácticas XML

La respuesta: No hay claves, pero hay recomendaciones:

- Se recomienda usar atributo cuando se vaya a categorizar información
- Cuando el valor sea único.

3.2. XML bien formado

- Un documento XML se dice “bien formado” (well-formed) cuando cumple las reglas descritas en la especificación oficial XML v1.0

3.2. XML bien formado

- Cómo puedo probar si un documento XML esta bien formado?

Uso un analizador sintáctico (parser)

Por ejemplo:

- Internet Explorer
- Aplicación para editar XML (por ejemplo XML Spy)
- Analizadores en linea (w3c)

3.3. Espacio de nombres

Responder a las preguntas:

- Puedo repetir el nombre de un elemento en cualquier documento XML?
- Quién decide el nombre del elemento?
- Cómo uso nombres estándares en los elementos para mejorar el intercambio de información?

3.3. Espacio de nombres

Colisión entre elementos:

```
<libro>  
<autor>Deepak Chopra</autor>  
<titulo>El sendero del  
Mago</titulo>  
<precio  
moneda="bolivares">30.000</pr  
ecio>  
</libro>
```

libro.xml

```
<cliente>  
<nombre>José Pérez</cliente>  
<titulo>Dr.</titulo>  
<email>perez@yahoo.com</em  
ail>  
<tarjeta>45440029292</tarjeta>  
</cliente>
```

cliente.xml

3.3. Espacio de nombres

Colisión entre elementos:

```
<orden>  
<autor>Deepak Chopra</autor>  
<titulo>El sendero del Mago</titulo>  
<precio moneda="bolivares">30.000</precio>  
<nombre>José Pérez</cliente>  
<titulo>Dr.</titulo>  
<email>perez@yahoo.com</email>  
<tarjeta>45440029292</tarjeta>  
</orden>
```

compra.xml

3.3. Espacio de nombres

XML namespaces

- Identifica la semántica de los elementos y atributos especialmente en el caso en el caso donde el documento tiene elementos con el mismo nombre pero diferente significado.
- Ampliamente usado en aplicaciones para asegurar la consistencia del significado de los nombres.

3.3. Espacio de nombres

```
<orden
xmlns:bk="http://www.net-standard.com/namespaces/books"
xmlns:cust="http://www.net-standard.com/namespaces/customer"
>
< bk:autor>Deepak Chopra</autor>
< bk:titulo>El sendero del Mago</titulo>
< bk:precio moneda="bolivares">30.000</precio>
<cust:nombre>José Pérez</cliente>
<cust:titulo>Dr.</titulo>
<cust:email>perez@yahoo.com</email>
<cust:tarjeta>45440029292</tarjeta>
</orden>
```

3.4. Espacio de nombres

- Se coloca en el momento de apertura del elemento que usa el namespace
- Los atributos no pertenecen al namespace del elemento. Hay que colocar el prefijo a los atributos, caso contrario se toma el de defecto.

PRÁCTICA #2

Usar el archivo **libro2.xml** que se encuentra en el directorio **tallerxml**, con el Internet Explorer y determinar si está “bien formado”. En caso contrario hacer los cambios necesarios en su definición según las reglas de construcción de XML version 1.0.

PRÁCTICA #2

<?xml version="2.0"?>

<libro>

<autor>Deepak Chopra

<titulo>El sendero del Mago</autor></titulo>

<isbn>950-15-1727</isbn>

<editorial>Harmany Book</editorial>

<sumario>En esta obra, Deepak Chopra, autor de varios libros que han ocupado los primeros puestos en las listas de ventas, nos muestra cómo debemos ... Por medio de historias como Gail & Jarret... </sumario>

<precio moneda="bolivares">30.000</precio>

<otro/>

</libro>

4 - Documentos XML válidos

1. Técnica de validación XML
2. DTD
3. XML Schemas
4. Lenguajes y Estándares XML

PRÁCTICA #3: Validando documentos XML
con DTD

4.1. Técnica de validación XML

- Qué se valida?

La estructura: nombres y valores de los elementos, atributos y entidades

- Para qué validar?

Consistencia de datos, compartir datos, uso de estándares.

4.1. Técnica de validación XML

Existen 2 formas de definir los elementos que contiene un documento XML a través de reglas gramaticales de los elementos, atributos y entidades:

- **DTD (Document Type Definition)**

Archivos con extensión .dtd

- **XML Schema**

Archivos con extensión .xsd

4.1. Técnica de validación XML

- Ambas formas nos permiten crear nuestro propio lenguaje de marcado.
- Ambas formas pueden residir en un archivo externo y ser compartidos por varios documentos XML.
- Un XML que se ajusta a cualquier técnica es un “XML válido”.
- Ambas técnicas son opcionales!

4.2. DTD

Un ejemplo de DTD: libro.dtd

```
<!ELEMENT libro (autor, titulo, isbn, editorial, sumario,  
    precio, otro?)>
```

```
<!ELEMENT autor (#PCDATA)>
```

```
<!ELEMENT titulo (#PCDATA)>
```

```
<!ELEMENT isbn (#PCDATA)>
```

```
<!ELEMENT editorial (#PCDATA)>
```

```
<!ELEMENT sumario (#PCDATA)>
```

```
<!ELEMENT precio (#PCDATA)>
```

```
<!ATTLIST precio moneda CDATA #REQUIRED>
```

```
<!ELEMENT otro (#PCDATA)>
```

4.2. DTD

- Uso Externo

```
<?xml version="1.0"?>
```

```
<!DOCTYPE libro SYSTEM "libro.dtd">
```

- Uso Interno

```
<?xml version="1.0"?>
```

```
<!ELEMENT libro (autor, titulo, isbn, editorial, sumario,  
    precio, otro?)>
```

```
<!ELEMENT autor (#PCDATA)>
```

...

```
<!ELEMENT otro (#PCDATA)>
```

```
<libro>
```

...

4.3. XML Schema

Un ejemplo de XML Schema: libro.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <element name="libro">
    <complexType>
      <sequence>
        <xsd:element name="autor" type="xsd:string"/>
        <xsd:element name="titulo" type="xsd:string"/>
        <xsd:element name="isbn" type="xsd:string"/>
        <xsd:element name="editorial" type="xsd:string"/>
        <xsd:element name="sumario" type="xsd:string"/>
        <xsd:element name="precio" type="xsd:string">
          <attribute name="moneda" type="string" use="required"/>
        </xsd:element>
        <xsd:element name="otro" type="xsd:string"/>
      </sequence>
    </complexType>
  </element>
</xsd:schema>
```

4.3. XML Schema

Un XML Schema es “similar” a un DTD, solo que:

- XML Schema usa sintaxis XML al contrario de los DTD
- Permite especificar los tipos y grupos de datos
- Son extensibles y tienen modularidad
- Usan namespace

4.3. XML Schema

- Uso Externo:

```
<?xml version="1.0"?>
```

```
<libro xmlns:xsi="http://www.w3.org/2001/XMLSchema-  
instance"  
xsi:noNamespaceSchemaLocation="libro.xsd">
```


4.4. Lenguajes y Estándares XML

- XHTML (eXtended HTML)
- WML (Wireless Mark-up Language) para dispositivos inalámbricos
- SVG (Scalable Vector Graphics) para producir imágenes
- RDF (Resource Definition Framework)
- VoiceXML
- SMIL Multimedia integrada

4.4. Lenguajes y Estándares XML

Lenguajes de Intercambio:

- ebXML - Comercio electrónico
- HL7 – Hospitales y Salud
- NewsML – Noticias
- RSS – Noticias
- SOAP
- DC (Dublin Core)

PRÁCTICA #3

En el directorio **tallerxml** se encuentra el parser de Microsoft **msxsl.exe**. Usar este parser para validar el archivo **libro3.xml** con el DTD **libro.dtd**.

> msxsl

> msxsl libro3.xml validar.xsl -v

5 - Consultando documentos XML

1. Tecnología para el manejo del XML
2. XPath
3. Xquery

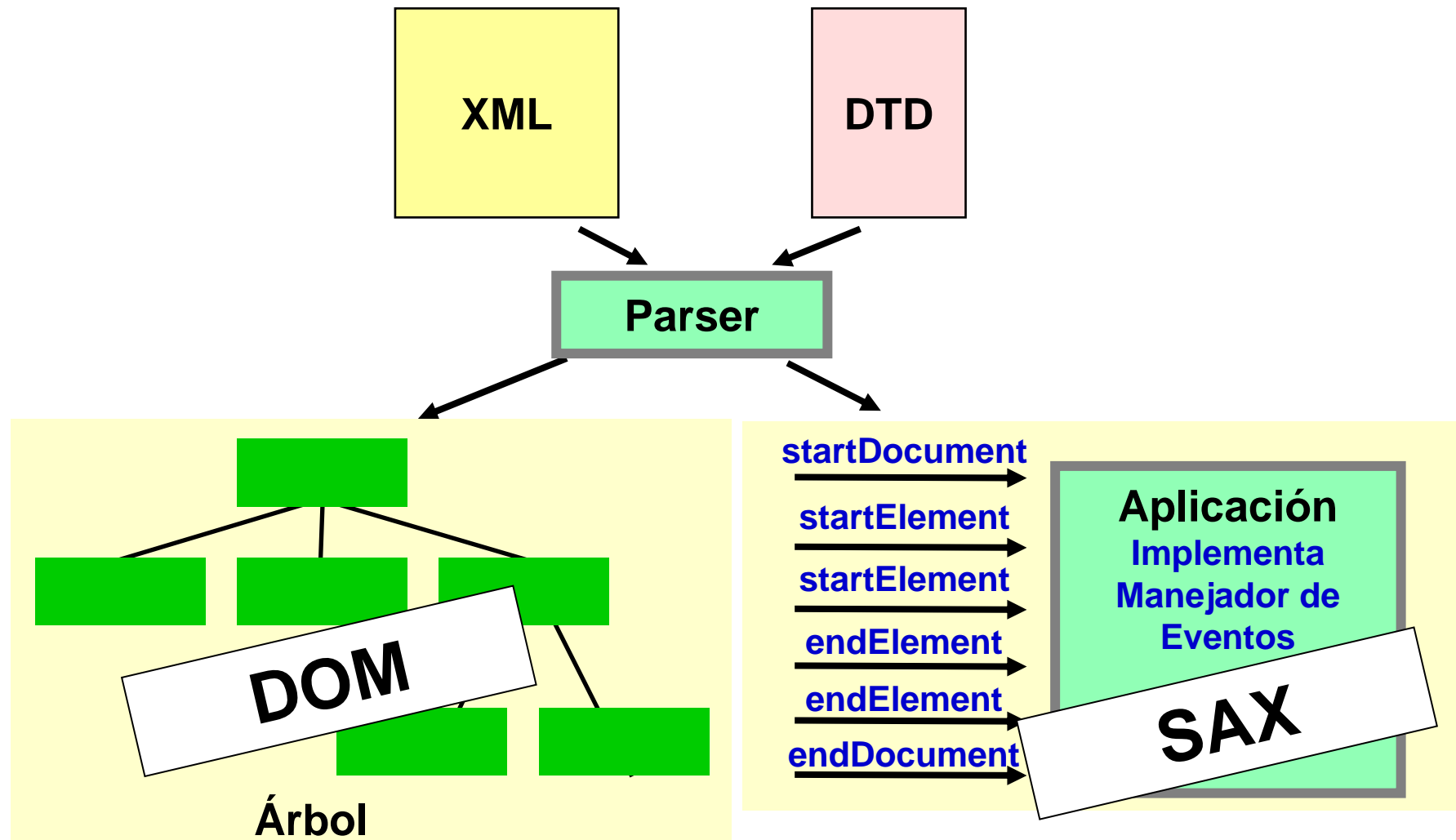
PRÁCTICA #4: Sentencias XPath

5.1. Tecnología para el manejo de XML

Se tienen 2 opciones “excluyentes” implementadas por el parser para manipular el contenido de un documento XML:

- Document Object Model (DOM)
- Simple API for XML (SAX)

5.1. Tecnología para el manejo de XML



5.1. DOM

- Acceso directo basado en la jerarquía
- Carga el árbol completo en memoria
- Analiza la validación y formación de cada nodo
- Métodos para navegación

5.1. SAX

- Acceso secuencial
- No se carga el árbol completo en memoria
- Modelo de análisis orientado a eventos
- Registro de eventos

5.2. XPath

- Localización y selección de elementos en el documento
- Navegación en el árbol del documento
- Recomendación W3C desde 1999
- <http://www.w3c.org/tR/xpath>

5.2. XPath

Especificación de patrones Xpath:

- / - Especifica el hijo inmediato. Puede referirse a la raíz
- // - Selecciona a cualquier profundidad de Árbol
- . - Selecciona el nodo contexto actual
- * - Selecciona todos los elementos en el nodo contexto actual

5.2. XPath

Especificación de patrones Xpath:

@ - Selecciona un atributo

@* - Selecciona todos los atributos en el **nodo contexto** actual

[] - Filtro condición de búsqueda

5.2. XPath

Nodo Contexto:

Nodo del árbol en donde está ubicado el cursor de la navegación durante el procesamiento de un documento XML.

Al inicio, por defecto, el nodo contexto es el nodo raíz.

5.2. XPath

Ejemplos de consultas Xpath:

<libro>

<autor>Deepak Chopra</autor>

<titulo>El sendero del Mago</titulo>

<isbn>950-15-1727</isbn>

<editorial>Harmany Book</editorial>

<sumario>... nos muestra cómo debemos ... Por medio de
historias como </sumario>

<precio moneda="bolivares">30.000</precio>

<otro/>

</libro>

5.2. XPath

Ejemplos de consultas Xpath:

- Título del libro: **`/libro/titulo`**
- Título del libro: **`//titulo`**
- Cuál es la Moneda del precio del libro?:
`/libro/precio/@moneda`
- Todos los datos del libro: **`/libro/*`**
- El libro cuyo precio se paga en bolivares: **`/libro[precio/@moneda = 'bolivares']`**

5.3. Xquery

Es una serie de especificaciones para consultas de XML aún en elaboración por el W3C:

- Consultas sobre uno o mas documentos XML (Base de datos XML).
- Consultas en sintaxis XML
- Consultas basadas en un estandar semejante a SQL.

PRÁCTICA #4

Probar las consultas Xpath del ejemplo en el documento libro2.xml cambiando el término “.” en la instrucción `<xsl:value-of select=“.”/>` por la consulta. Usar el siguiente llamado del parser msxsl:

```
> msxsl libro4.xml xpath.xsl
```


6- Transformando XML

1. Técnicas de presentación XML
2. CSS
3. XSLT, XSL, XSL-FO

PRÁCTICA #5: Demostraciones del instructor de Transformaciones con un parser XSL

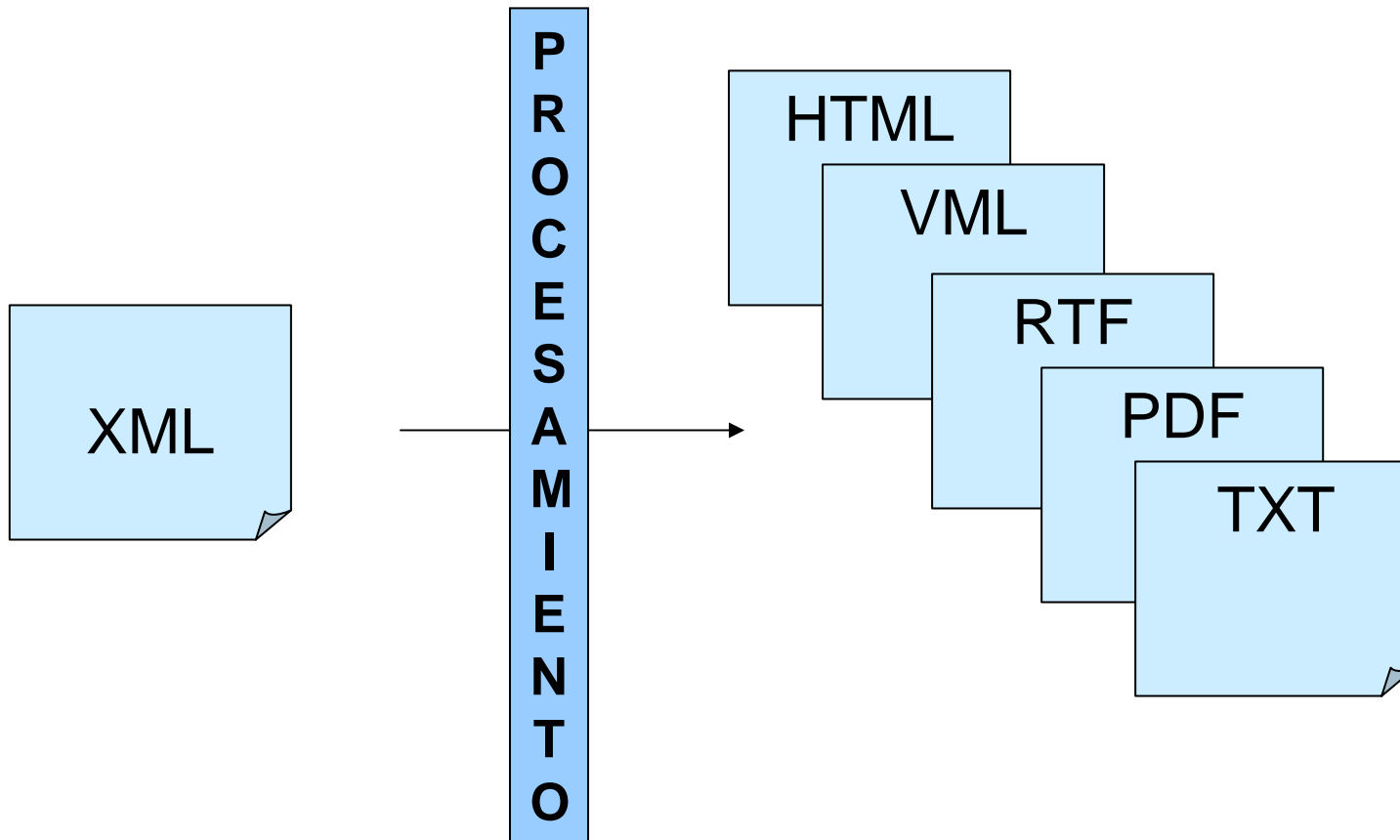
6.1. Técnicas de Presentación

Se tienen 2 técnicas de presentación del contenido de un documento XML:

- CSS (Cascading Style Sheets)
- XSL (XML Stylesheet Language)

6.1. Técnicas de Presentación

Muchas formas de presentación a partir de un documento XML:



6.2. CSS

- CSS es la forma más popular de dar formato a archivo HTML
- CSS también se usa para dar formato a archivos XML
- El formato que permite obtener es sólo para interfaces Web (HTML)

6.2. CSS

CSS permite definir a la salida HTML:

- Propiedades de fuente
- Propiedades de color y fondo
- Propiedades de texto
- Propiedades de posición
- Listas y tablas
- Etc.

6.2. CSS

Características:

- El tratamiento del documento XML es secuencial
- Se procesa toda la data del documento XML
- El procesamiento se realiza en el cliente (Web Browser)

6.2. CSS

Sentencias básicas:

```
selector { propiedad: valor;  
          ... propiedad: valor;}
```

Por ejemplo:

```
libro autor { color:navy;  
              font-family:Brush Script,Fantasy;  
              font-size:18pt;  
              display:block; }
```

6.2. CSS

Cómo relaciono el archivo CSS con el documento XML?

```
<?xml version="1.0">  
<?xml-stylesheet type="text/css"  
href="libro.css"?>  
<libro>  
<autor>Deepak Chopra</autor>  
<titulo>El sendero del Mago</titulo>  
<precio  
moneda="bolivares">30.000</precio>  
</libro>
```

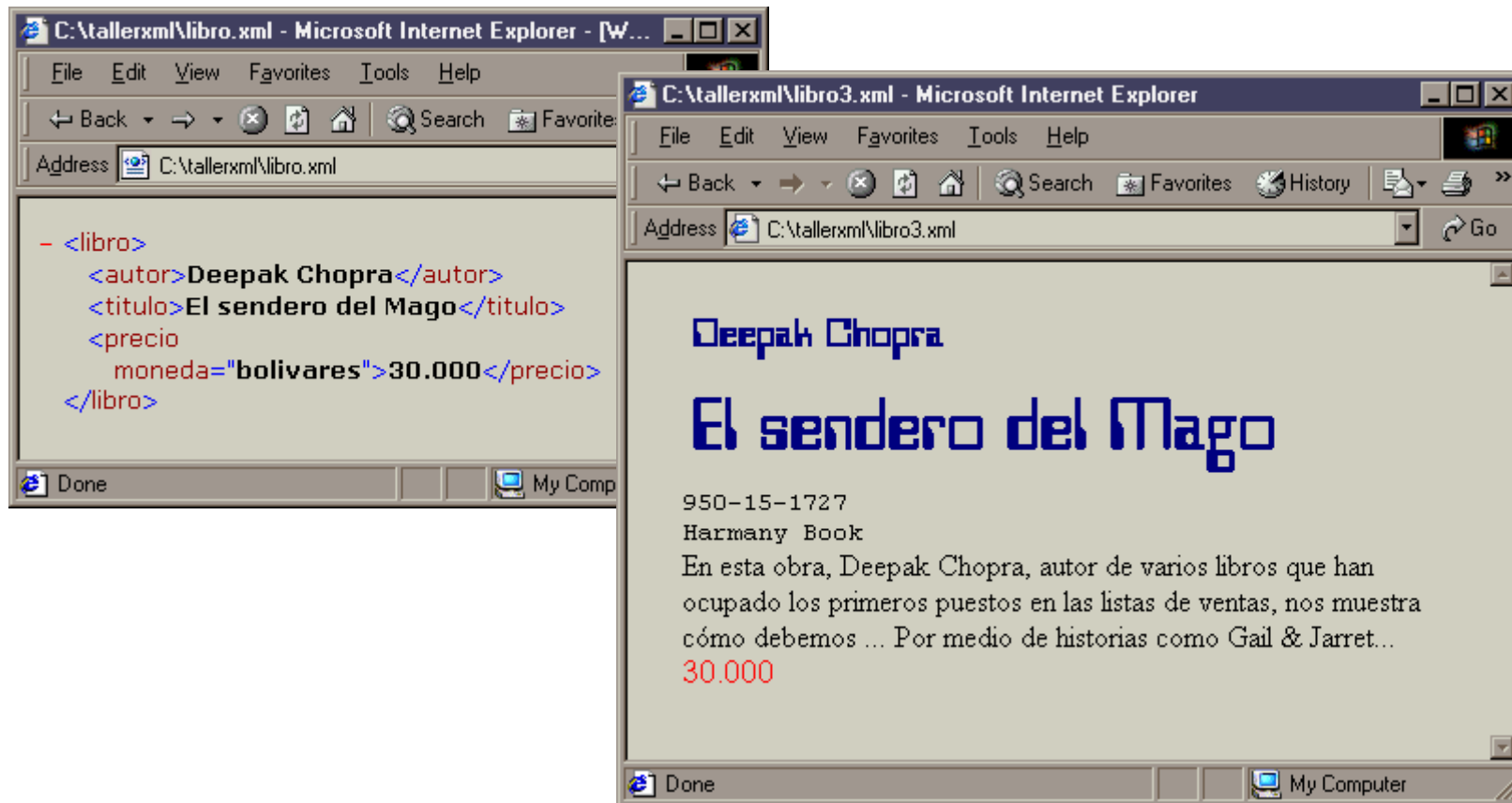
libro.xml

```
libro { display:block;  
        font-family:Arial;  
        margin:0.2in;  
        width:400px  
        }  
libro titulo { width:400px;  
               color:navy;  
               ...
```

libro.css

6.2. CSS

XML con CSS visto en un Web Browser

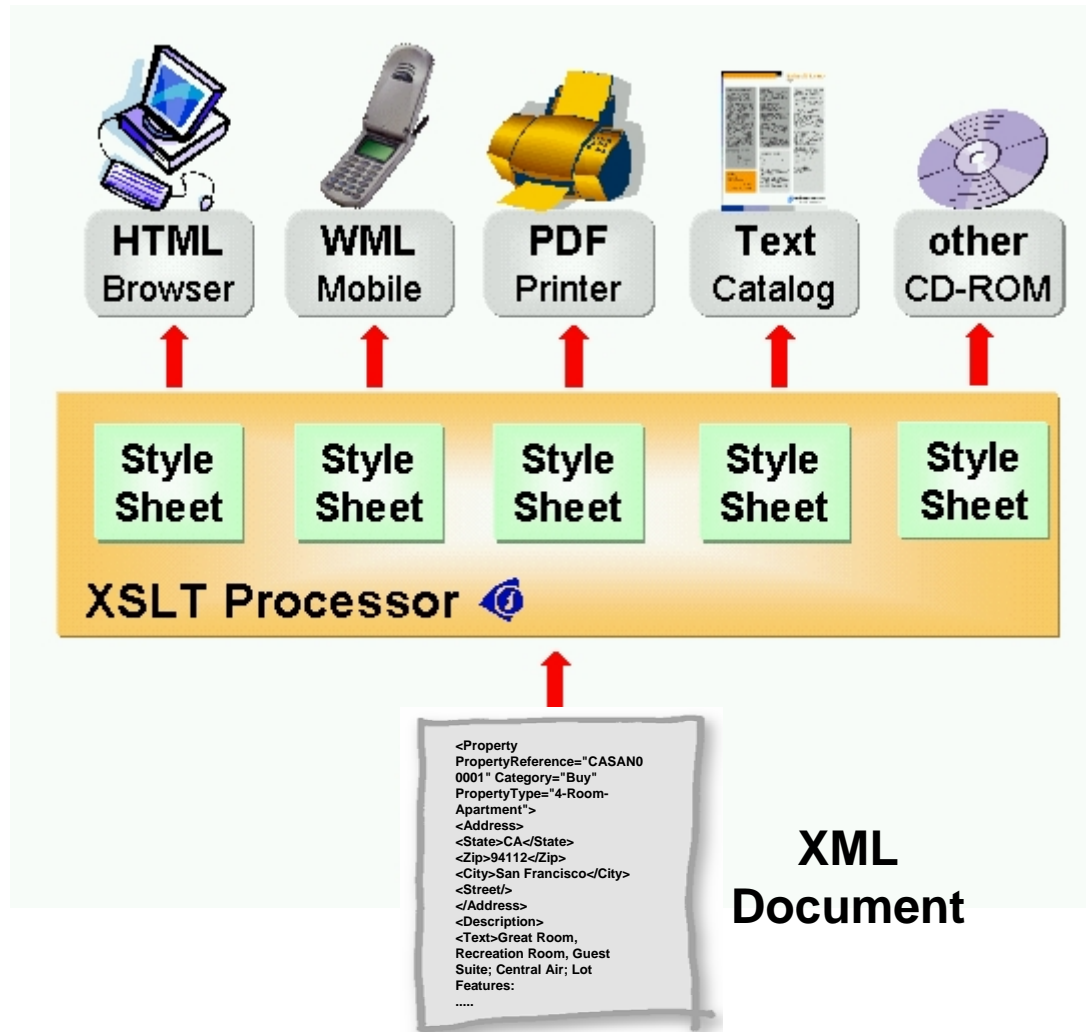


6.3. XSL

XSL es un lenguaje que nos permite definir diferentes tipos de presentaciones y procesamientos para un documento XML mostrando diferentes formatos (HTML, TXT, PDF, XML, PostScript, etc.).

Recomendación W3C 1999

6.3. XSL



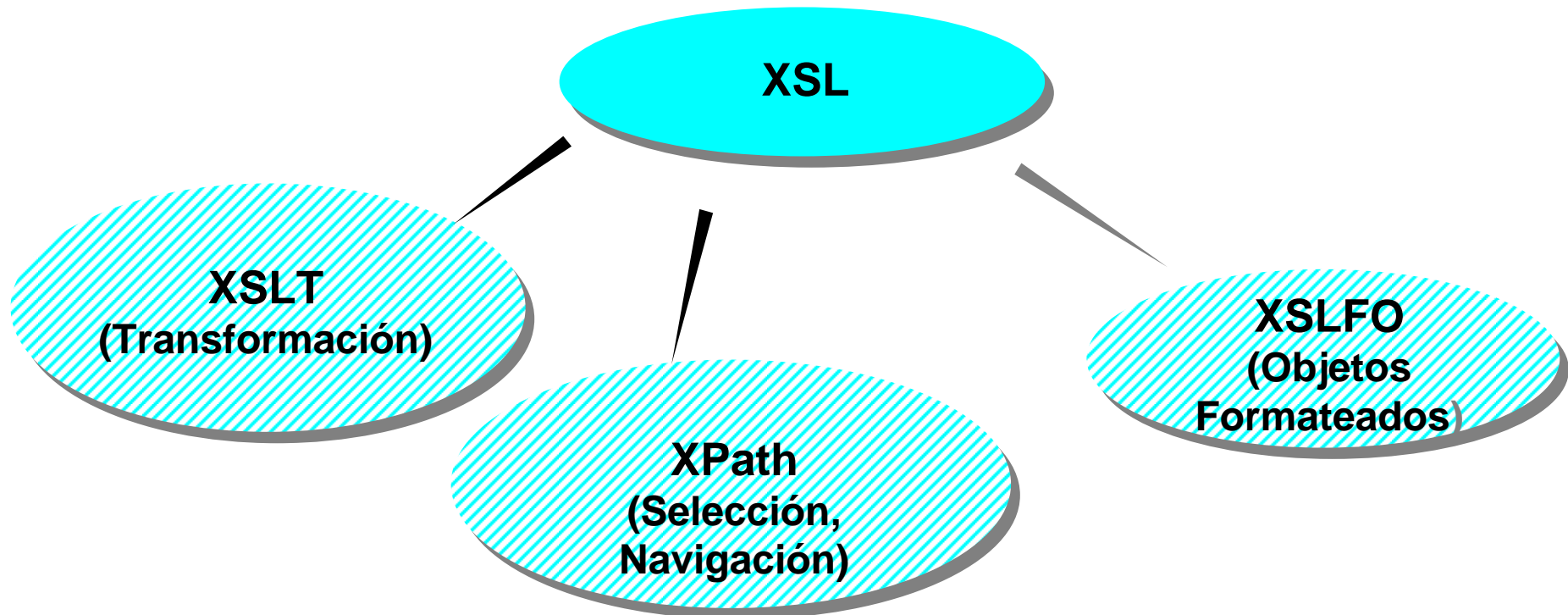
6.3. XSL

- XSL define las transformaciones del documento XML
- Una hoja de estilo XSL es una serie de reglas que determinan cómo va a ocurrir la transformación. Cada regla se compone de un patrón [pattern] y una acción o plantilla [template]

6.3. XSLT

- XSLT (XSL Transformation)

$XSL = XSLT + XPath + XSLFO$



6.3. XSL

Una hoja de estilo XSLT es un archivo XML con extensión .xsl

El elemento raíz es: `<xsl:stylesheet>`

```
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
</xsl:stylesheet>
```

ó

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
</xsl:stylesheet>
```

6.3. XSL

Elementos básicos de XSLT:

- Para definir patrones de un conjunto de nodos:

```
<xsl:template match="pattern">
```

```
</xsl:template>
```

- Para insertar el valor del nodo seleccionado:

```
<xsl:value-of select="pattern" />
```

6.3. XSL

Ejemplo de una hoja de estilo XSL:

```
<?xml version="1.0" ?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html"/>
<xsl:template match="/">
    <html><body>
        <B><xsl:value-of select="//titulo"/></B>
    </body></html>
</xsl:template>
</xsl:stylesheet>
```


6.2. XSL

Cómo relaciono el archivo XSL con el documento XML?

```
<?xml version="1.0">  
<?xml-stylesheet type="text/xsl"  
href="libro.xsl"?>  
  
<libro>  
  
<autor>Deepak Chopra</autor>  
  
<titulo>El sendero del Mago</titulo>  
  
<precio  
moneda="bolivares">30.000</precio>  
  
</libro>
```

libro.xml

```
<?xml version="1.0" ?>  
  
<xsl:stylesheet version="1.0"  
xmlns:xsl="http://www.w3.org/TR/WD-  
xsl">  
  
<xsl:template match="/">  
  
  <html><body>  
  
    <H2><xsl:value-of  
select="libro/titulo" /></H2> ...  
  
</xsl:stylesheet>
```

libro.xsl

6.3. XSL

XML con XSL visto en un Web Browser



PRÁCTICA #5

Demostración del instructor:

- XML con CSS (libro5-1.xml y libro.css)
- XML con XSL (libro5-2.xml y libro.xsl)

7 - Uso de XML en Bases de Información

1. Integrando XML en aplicaciones
2. Generando salidas XML desde una Base de Información (BI)
3. Uso de XML como formato de intercambio (RSS, MARC-XML, OAI, etc)

PRÁCTICA #5: XML en aplicaciones y Bases de Información

Advertencia!

XML y sus tecnología pueden parecer muy útiles y potentes!...

Pero ... debemos usar la herramienta más adecuada en cada caso

No por tener un martillo todo se convierte en clavo...

GRACIAS!